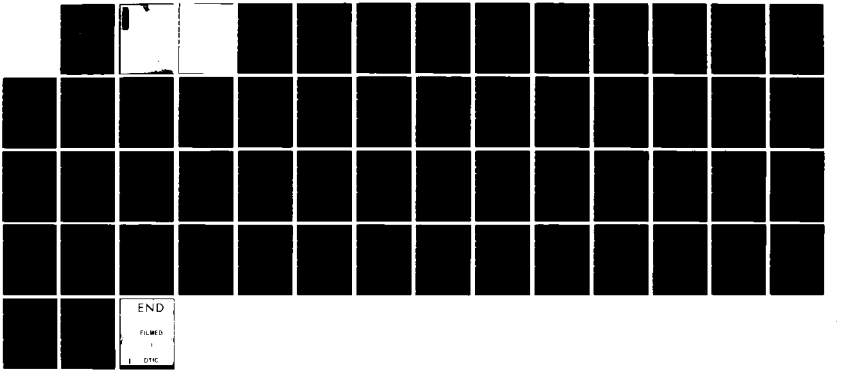
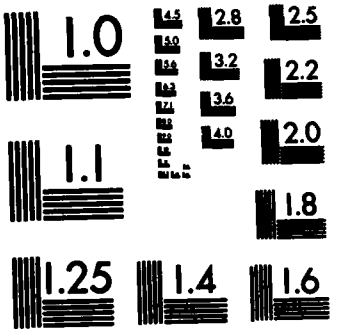
AD-A123 419 RESEARCH ON ALGORITHMS FOR ADAPTIVE ARRAY ANTENNAS(U) 1/1
STANFORD UNIV CA DEPT OF ELECTRICAL ENGINEERING
B WIDROW ET AL. NOV 82 RADC-TR-82-297 F30602-80-C-0046
UNCLASSIFIED F/G 9/5 NL





UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-82-297	2. GOVT ACCESSION NO. AD-A123419	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RESEARCH ON ALGORITHMS FOR ADAPTIVE ARRAY ANTENNAS		5. TYPE OF REPORT & PERIOD COVERED Interim Report Feb 81 - Feb 82 6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) B. Widrow R. Gooch T. AuTruong		8. CONTRACT OR GRANT NUMBER(s) F30602-80-C-0046
9. PERFORMING ORGANIZATION NAME AND ADDRESS Stanford University Dept of Electrical Engineering Stanford CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2305J805
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (DCCD) Griffiss AFB NY 13441		12. REPORT DATE November 1982 13. NUMBER OF PAGES 53
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: John A. Graniero (DCCD)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive Systems	Adaptive Arrays
Adaptive Filters	Null Steering
Adaptive Antennas	IIR Filters
	Lattice Filters

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report demonstrates a method for implementing a wide-band constrained array processor using filters possessing both poles and zeros. The idea of constrained array processing is introduced and an optimal array weighting formula is derived. This optimal weighting is meant to cancel m-1 wide-band interference sources incident on an m-element array with a constrained look direction. From the optimal weighting formula, it becomes evident that the use of filters possessing both poles and zeros

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

to perform the frequency-dependent weighting has the potential for improving the wide-band nulling capability of the array. A novel method for adapting the pole-zero array filters in tap-delay-line form using the LMS or RLS algorithm is presented. The issues of look-direction signal bias and stability of the inverse filter are addressed.

A method for adaptively implementing the pole-zero array filters in lattice form is detailed. It is shown that the use of an adaptive lattice filter offers a considerable improvement in the algorithm's speed of convergence.

An extension of the pole-zero lattice predictor to multi-element antenna processing is presented. It is shown that the computational complexity of such a multivariable filter grows as the cube of the dimension of the input vector and directly with the filter order.

TABLE OF CONTENTS

I. INTRODUCTION

This report demonstrates a method for implementing a wide-band constrained array processor using filters possessing both poles and zeros. In section II, the idea of constrained array processing is introduced and an optimal array weighting formula is derived. This optimal weighting is meant to cancel $m-1$ wideband interference sources incident on an m-element array with a constrained look direction. From the optimal weighting formula, it becomes evident that the use of filters possessing both poles and zeros to perform the frequency-dependent weighting has the potential for improving the wideband nulling capability of the array. A novel method for adapting the pole-zero array filters in tap-delay-line form using the LMS or RLS algorithm is presented. In the remaining part of this section, the issues of look-direction signal bias and stability of the inverse filter are addressed.

In subsequent sections, a method for adaptively implementing the pole-zero array filters in *lattice form* is detailed. It is shown that the use of an adaptive lattice filter offers a considerable improvement in the algorithm's speed of convergence. Convergence rate is an important characteristic of an adaptive algorithm. A rapidly converging algorithm eliminates its dependence on initial conditions and improves its ability to track time-varying scenarios. In adaptive array processing, gradient-descent algorithms are widely used because of their simplicity of implementation. When the input signal autocorrelation matrix **R** is "close" to identity, the convergence rate of a gradient-descent algorithm is as fast as a Newton-descent algorithm. The convergence rate of a gradient-descent algorithm is, however, dependent upon the eigenvalue disparity of the **R** matrix. On the other hand, lattice algorithms are known to be relatively insensitive to eigenvalue spread. This insensitivity, along with an otherwise fast convergence capability, should reduce any concern about the computational complexity of the lattice.

In section III, we present the scalar lattice predictor in the context of speech processing where it was first introduced. We show that the inverse of a lattice predictor can be implemented in lattice form and that its stability can be easily checked. We also present the joint-process lattice used for estimating one signal based on observations of another signal. Simulations of the

algorithm are presented to show its performance under a variety of different input eigenvalue spreads.

We continue in section IV by presenting the 2-channel lattice predictor and show how it can be used for equation error pole-zero filtering by embedding the primary and auxiliary inputs into a multi-channel input. Simulations showing the 2-C lattice predictor performance when there is an eigenvalue spread are presented.

In section V, we apply the lattice predictor to a two-element pole-zero array processing problem. The question of forming an inverse filter to restore the true output error is solved via the computation of an equivalent scalar lattice predictor. To complete this section, antenna processing simulations using the pole-zero generalized sidelobe canceler are presented.

In section VI we present an extension of the pole-zero lattice predictor to multi-element antenna processing. We point out that the computational complexity of such a multivariable filter grows as the cube of the dimension of the input vector and directly with the filter order.

And last, in section VII we present our conclusions and suggestions for future research.

II. WIDE-BAND ADAPTIVE ARRAY PROCESSING USING POLES AND ZEROS

2.1 Introduction

Conventional adaptive array processing is accomplished by linearly combining the outputs of tap delay lines attached to each sensor of an array. This type of processing can be interpreted as using an "all-zero" filter at each sensor to generate a frequency-dependent magnitude and phase shift (weighting) over the desired array bandwidth. In this section of the report we present a new method for performing the frequency-dependent array weighting using filters possessing both poles and zeros. Adaptation of these filters is based on "equation error" rather than on the usual "output error". The method is shown to substantially improve the wideband interference nulling capabilities of the array, providing sharper and deeper nulls. It is expected to be useful in many seismic, acoustic, and electromagnetic array processing applications.

The sequence of this section is as follows. First we introduce the concept of linearly constrained array processing. Then we derive the optimal array weighting required to eliminate $m-1$ interference sources incident on an m-element array with a constrained look direction. The form of this optimal weighting is used as motivation for developing an array processor based on pole-zero filters. Next, through simulations we show how a combination of both poles and zeros can substantially improve the approximation of the optimal weighting compared to that obtained using the conventional all-zero method. Finally, the last part of this section addresses the issues of look-direction signal bias and inverse filter stability which arise because the true output error is not being minimized.

2.2 Constrained Array Processing

A constrained array processor such as the one proposed by Frost [1], operates in an environment where a single desired signal is incident upon an array from a known direction and several interference signals are incident from unknown directions. The objective of a Frost array processor is to minimize the response of the array in the direction of the interference signals while leaving the response in the look direction unaltered. A block diagram of the Frost array is shown in

Figure 2.1. Throughout this paper Z-transform notation is used to indicate a fixed linear digital filter. To convert from the Z-domain into a function of frequency, the substitution $z=e^{j\omega}$ is made. The look-direction signal is assumed to be perpendicular to the array axis while interference signals are incident from other unknown directions. If the look direction is not perpendicular to the array axis, alignment filters usually in the form of steering delays can be added to cause the look-direction signal to appear in time coincidence at the output of each delay. Such a system is referred to as a signal-aligned array [2]. After alignment, the signals are passed through linear filters and then summed to form the array output. Because the look-direction signal appears identically at the input to each array filter, it will experience a response through the array determined by the sum of all the array filters. In order to maintain an undistorted response in the look-direction, this sum is constrained to be unity. Subject to the constraint, the linear filters are adjusted to minimize the output power of the array. In this manner, interference signals are eliminated without distorting the look-direction signal. We now determine the optimal *frequency-dependent weighting required* to cancel $m-1$ broadband directional interference sources incident upon an m-element array. The analysis assumes ideal propagation conditions and negligible thermal noise. Let the Z-transform of the look-direction signal be $S(z)$ and the transform of the l^{th} interference signal be $R_l(z)$. The look-direction signal appears identically at all sensors of the array and is desired to appear undistorted at the array output. Maintaining zero distortion in the look-direction requires that

$$S(z) \sum_{k=1}^{m} W_k(z) = S(z) \quad . \tag{2.1}$$

Cancelling S(z) from both sides of this equation leaves,

$$\sum_{k=1}^{m} W_k(z) = 1 \quad , \tag{2.2}$$

the look-direction constraint. For simplicity we assume a uniformly spaced linear array. Then each interference signal will experience a propagation delay, Δ_l, between adjacent sensors that is determined by the inter-sensor spacing, the speed of propagation, and the arrival angle. Com-

plete cancellation of all interference signals requires that

$$R_l(z) \sum_{k=1}^{m} W_k(z)\, z^{-(k-1)\Delta_l} = 0 \quad \text{for } l = 1,2,\ldots,m-1 \quad . \tag{2.3}$$

Equations (2.1) and (2.3) can be combined and written in matrix form as,

$$\begin{bmatrix} S(z) & & & \\ & R_1(z) & & 0 \\ & & \ddots & \\ & 0 & & R_{m-1}(z) \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & z^{-\Delta_1} & \cdots & z^{-(m-1)\Delta_1} \\ \vdots & \vdots & & \vdots \\ 1 & z^{-\Delta_{m-1}} & \cdots & z^{-(m-1)\Delta_{m-1}} \end{bmatrix} \cdot \begin{bmatrix} W_1(z) \\ W_2(z) \\ \vdots \\ W_m(z) \end{bmatrix} = \begin{bmatrix} S(z) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.4}$$

By induction, it can be shown that the solution to this set of linear equations is,

$$\begin{bmatrix} W_m(z) \\ W_{m-1}(z) \\ \vdots \\ W_1(z) \end{bmatrix} = \left[\prod_{k=1}^{m-1} \frac{1}{(1 - z^{-\Delta_k})} \right] \begin{bmatrix} 1 \\ -\sum \textit{ all single delays} \\ \vdots \\ (-1)^{m-1} z^{-(\Delta_1 + \cdots + \Delta_{m-1})} \end{bmatrix} . \tag{2.5}$$

Use of this frequency-dependent weighting in a linear uniformly spaced array will cause all of the jammers to be completely nulled regardless of bandwidth. However, since the delays are usually fractional and the frequency resonances caused by the poles are infinite, an exact rational realization of this weighting is not possible. Instead, one must be content with a rational approximation that is made over a selected bandwidth. Array performance using tapped delay lines operating over a wide bandwidth has received some attention in the literature. Two key papers are those by Compton [3] and more recently by Mayhan and Simmons [4].

To gain insight into the frequency response of the optimal weighting, consider the 2-element single jammer case. When $m=2$, the optimal weight transfer functions are,

$$W_1(z) = \frac{-z^{-\Delta}}{1 - z^{-\Delta}} \tag{2.6a}$$

and

$$W_2(z) = \frac{1}{1 - z^{-\Delta}} \quad . \tag{2.6b}$$

Figure 2.2 shows the frequency response of the optimal weights. The relevant frequency range occurs with ω between 0 and π and typical values of Δ range between + 2 and –2 depending upon the difference in angle between the interference direction and the look direction. The infinite resonance at zero frequency indicates that it is physically impossible to maintain both unity d.c. gain in the look direction and a d.c. null in the interference direction. Also as the interference direction approaches the look direction, Δ becomes small and the frequency scale of the plot expands. This causes the zero frequency resonance to dominate the 0 to π frequency range and makes approximation of the optimal weights more difficult.

It is informative to compare the set of optimal weight equations given, in (2.4) with the well-known Wiener equations. When considering the effect of thermal noise on the array output, it turns out that under certain circumstances, the solution to eqs. (2.4) will perfectly null out the directional interference sources but at the expense of blowing up the thermal noise. Thus, while the signal-to-interference ratio is improved, there may be a substantial degradation in the signal-to-thermal noise ratio. For this reason, it is important to consider the effect of thermal noise in order to find a compromise between interference nulling and thermal noise suppression. This is essentially what Wiener theory does in minimizing the overall array output power subject to a constraint. It can easily be shown using Lagrange multipliers that the weights which minimize the mean square output of the array subject to a unity-gain look-direction constraint satisfy the following equations [1],

$$\mathbf{R}_{xx}(z)\cdot\mathbf{W}(z) = \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \Big/ \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \cdot \mathbf{R}_{xx}^{-1} \cdot \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \tag{2.7}$$

The mxm matrix $\mathbf{R}_{xx}(z)$ is the spectral covariance matrix of the array signals after alignment and $\mathbf{W}(z)$ is the vector of the optimal frequency-dependent weights.

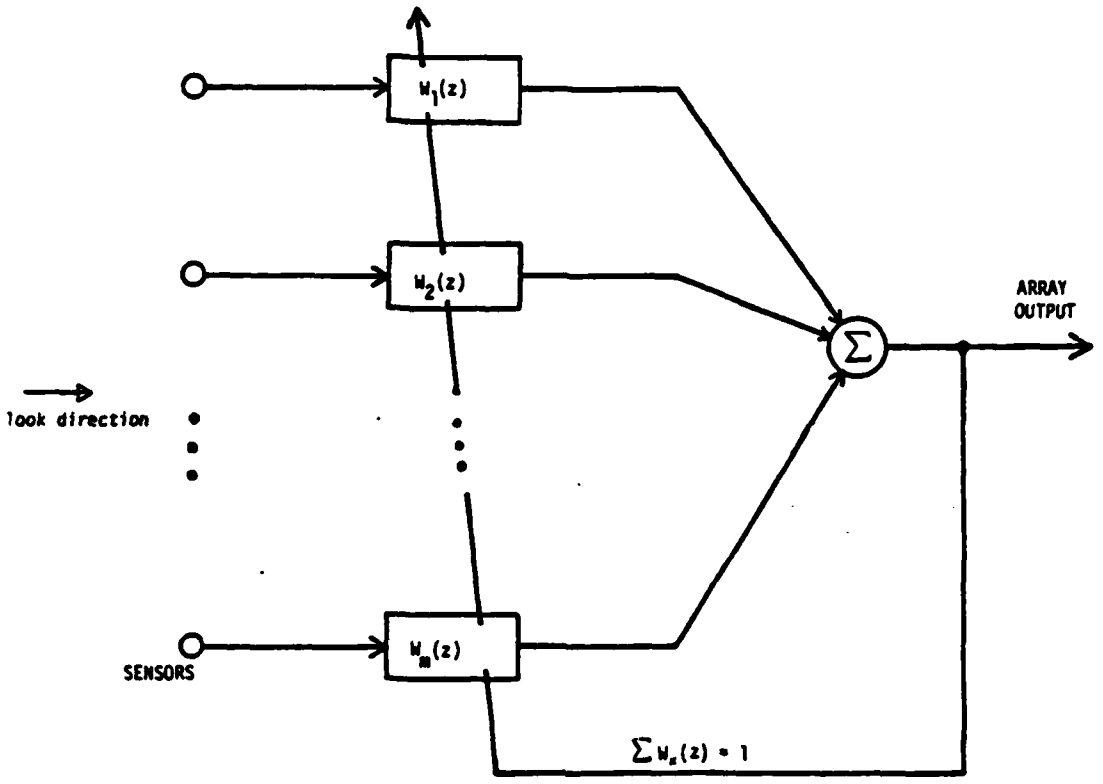


Figure 2.1 Block diagram of a Frost array with unity gain look-direction constraint.

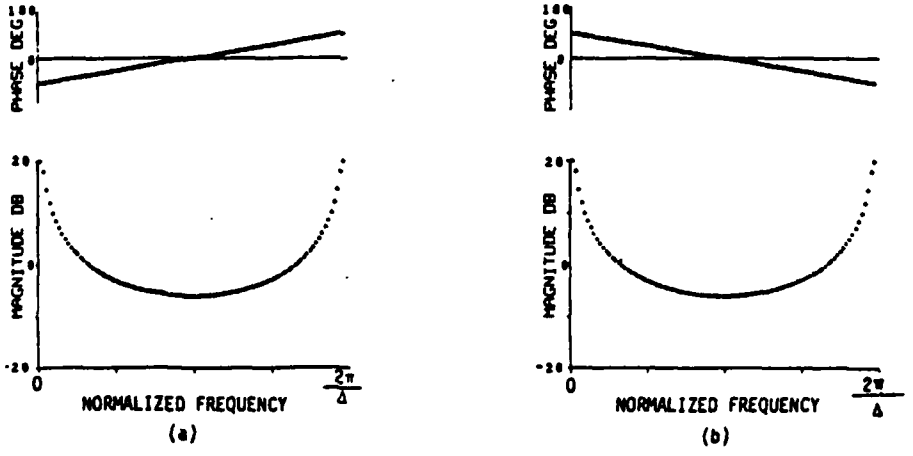


Figure 2.2 Frequency response of optimal array weights for a two-element Frost array with a single interference source incident on the array. The jammer arrival angle and sampling rate determine the intersensor delay Δ. (a) Sensor 1 (b) Sensor 2.

To connect this set of equations to the set given in (2.4), let (2.4) be more concisely written as,

$$\mathbf{A}(z)\cdot\mathbf{W}(z) = \begin{bmatrix} S(z) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} . \tag{2.8}$$

Then multiply both sides of (8) by $\mathbf{A}^T(z^{-1})$ to give,

$$\mathbf{A}^T(z^{-1})\mathbf{A}(z)\cdot\mathbf{W}(z) = \mathbf{A}^T(z^{-1})\cdot\begin{bmatrix} S(z) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} . \tag{2.9}$$

By interpreting $|S(z)|^2$ and $|R_l(z)|^2$ as the power spectra of stationary stochastic signals, the matrix $\mathbf{A}^T(z^{-1})\cdot\mathbf{A}(z)$ can be interpreted as the spectral covariance matrix,

$$\mathbf{A}^T(z^{-1})\cdot\mathbf{A}(z) = \mathbf{R}_{ss}(z) . \tag{2.10}$$

Also, it can easily be seen that,

$$\mathbf{A}^T(z^{-1})\cdot\begin{bmatrix} S(z) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} = |S(z)|^2\cdot\begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} . \tag{2.11}$$

Therefore eq. (2.9) can be written as

$$\mathbf{R}_{ss}(z)\cdot\mathbf{W}(z) = |S(z)|^2\cdot\begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} . \tag{2.12}$$

Since eqs. (2.4) were assumed to be satisfied, it now follows that

$$[\,1\;1\;\ldots\;1\,]\cdot\mathbf{R}_{ss}^{-1}(z)\cdot\begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ 1 \end{bmatrix} = \frac{1}{|S(z)|^2} . \tag{2.13}$$

Thus with no thermal noise, eq. (2.4) is equivalent to the Wiener solution given in eq. (2.7).

2.3 Alternate Implementation of a Frost array processor

To simplify the adaptivity issue, we introduce an alternate constrained array processing structure first discussed by Applebaum and Chapman [5] and more recently shown to be equivalent to a Frost array by Griffiths and Jim [6]. A block diagram of this structure is shown in Fig. 2.3. The basic idea is to remove the look-direction signal from the adaptive process x-input. This makes it impossible for the adaptive process to cancel signal components from the array output by subtraction. Interference signals appear in both the x and d inputs and can therefore be canceled from the output. Transformation between the two array processors is readily seen to be,

$$\begin{bmatrix} W_1(z) \\ W_2(z) \\ \cdot \\ \cdot \\ \cdot \\ W_m(z) \end{bmatrix} = \frac{1}{m} \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & & & & \\ -1 & 1 & 0 & & \\ & -1 & \cdot & & \\ & & \cdot & \cdot & \\ & 0 & & \cdot & 1 \\ & & & & -1 \end{bmatrix} \cdot \begin{bmatrix} H_1(z) \\ H_2(z) \\ \cdot \\ \cdot \\ \cdot \\ H_{m-1}(z) \end{bmatrix} \tag{2.14}$$

For the $m=2$ case, this set of equations becomes,

$$\begin{aligned} W_1(z) &= 1/2 - H(z) \\ W_2(z) &= 1/2 + H(z) \quad . \end{aligned} \tag{2.15}$$

Solving for $H(z)$ gives,

$$H(z) = 1/2 \quad [W_2(z) - W_1(z)] \quad , \tag{2.16}$$

and the optimal filter formulas of Eq (2.6) reduce to,

$$H(z) = 1/2 \quad \frac{1 + z^{-\Delta}}{1 - z^{-\Delta}} \quad . \tag{2.17}$$

2.4 Simulation of a two-element Frost array processor

Figure 2.4 compares the frequency response of the converged Frost array filters with the frequency response of the optimal weighting. The interference signal had a 100% relative bandwidth

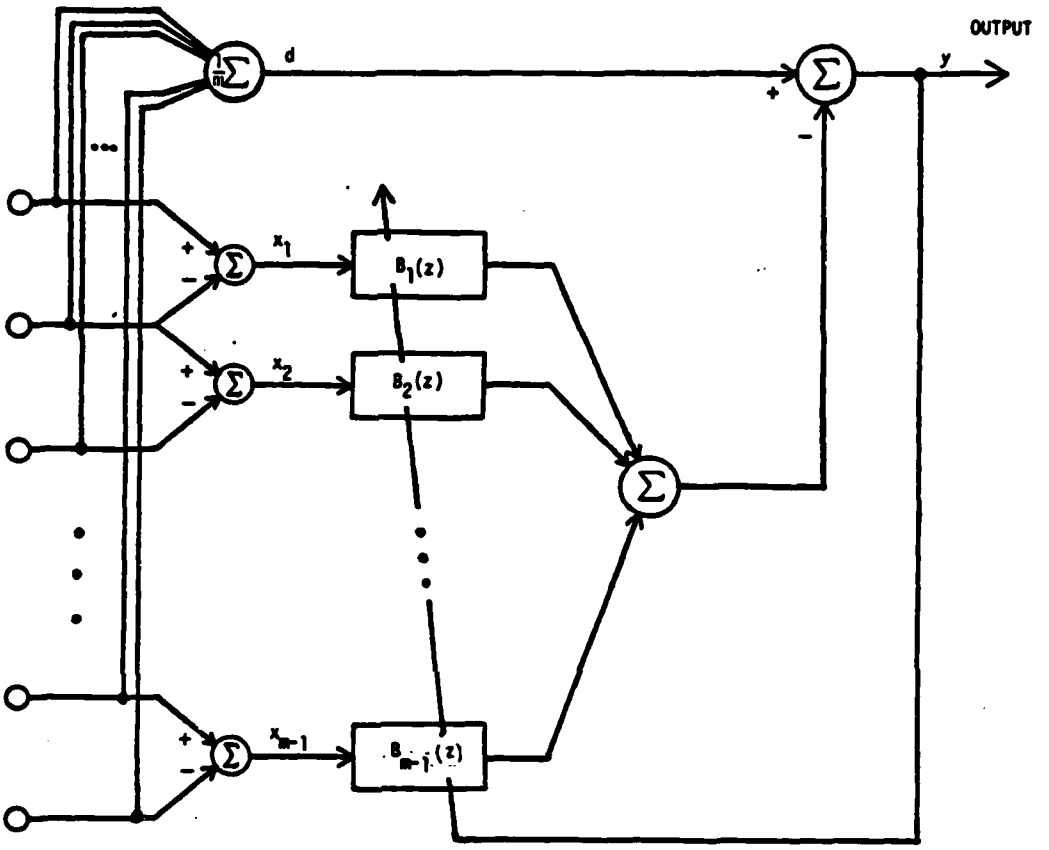


Figure 2.3 Alternate realization of a Frost array using unconstrained adaptation.

and an arrival angle of 45 degrees. The array consisted of two elements and the array filters each had 4 zeros. The cross-hatched dark lines indicate the optimal frequency response plotted over the bandwidth of the interference. The continuous curve represents the frequency response of the 4-zero Frost array filter. If the relative bandwidth were small enough, the optimal frequency response would need to be matched in magnitude and phase at only a single frequency. Such a response could easily be realized using one complex weight as is the case in most narrowband adaptive array systems. In this paper, however, we are concerned with performance over a wide bandwidth. Figure 2.5 shows the broadband antenna pattern formed by averaging narrowband patterns over the frequency range of the interference signal. The average gain in the direction of the interference is 20 dB below the signal. As more weights are added to the array filters, approximation of the optimal weight frequency response will become more accurate and the null depths will become deeper.

2.5 A pole-zero Frost array processor

Aside from adding more taps to the array filters, another way to improve the array performance is to use filters possessing both poles and zeros. In Fig. 2.2, we see that the optimal weight frequency response contains infinite resonances at the frequencies $n \cdot \pi/\Delta$. Such resonances are better approximated with filters having both poles and zeros than with zeros alone. In the next section, we introduce a method for adapting an array processor which has array filters possessing both poles and zeros.

Development of an adaptive algorithm to adjust combined pole-zero filters has been the subject of much research, but has met with only limited success. The reason for this is that the mean square error function associated with the pole-zero filter is nonlinear and possesses local minima. This complicates the problem of finding the global minimum of the mean square error. This problem does not arise in an all-zero adaptive filter since the mean square error function is quadratic in the weights and the unique minimum can be found by solving a set of linear equations.

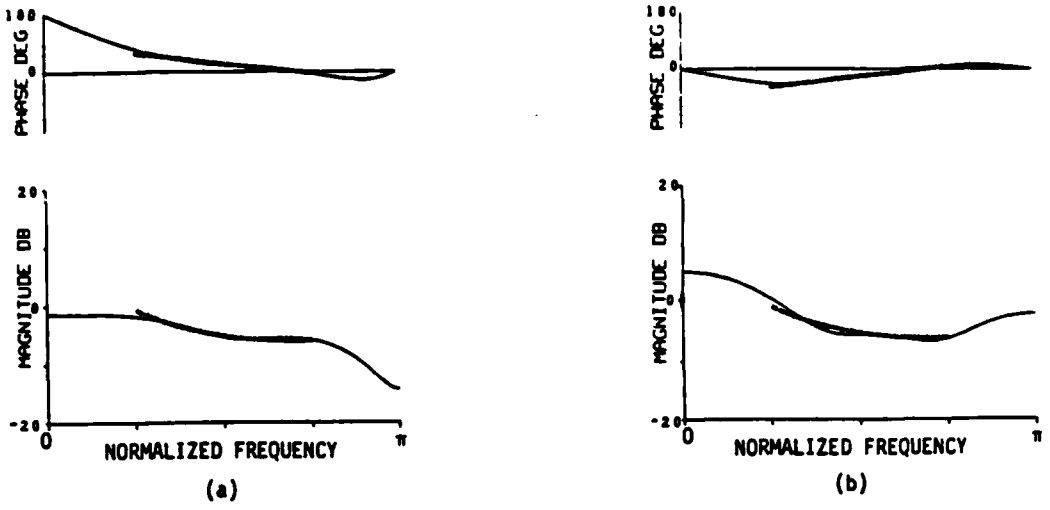


Figure 2.4 Frequency response of optimal array weights (cross-hatched marks) compared to 5 weight approximation from a Frost array. (a) Sensor 1 (b) Sensor 2.

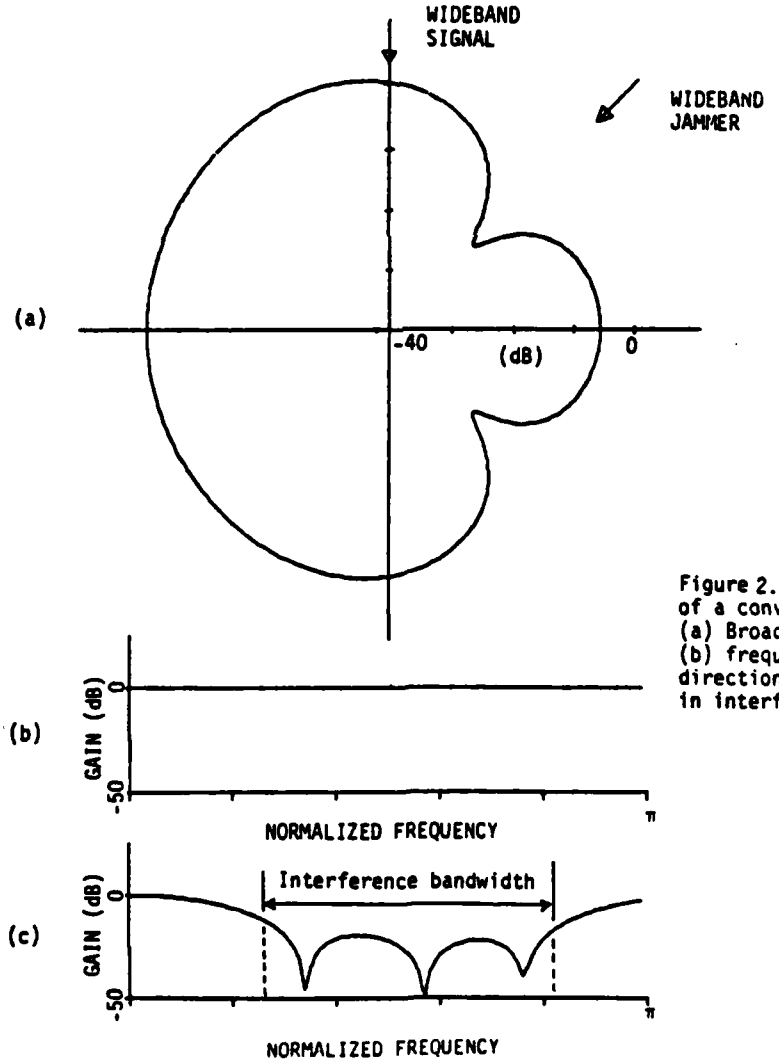


Figure 2.5 Nulling performance of a converged Frost array. (a) Broadband antenna pattern, (b) frequency response in signal direction, (c) frequency response in interference direction.

Our approach bypasses many of the problems usually associated with pole-zero adaptive filters. It involves a reformulation of the error function in a manner which allows the polynomial associated with the poles of the adaptive filter to be adapted in an all-zero form. This restores the quadratic nature of the minimization problem to that encountered in the all-zero case and allows easy adaptation using Least Mean Square Algorithm (LMS), the Recursive Least Squares Algorithm (RLS), the sample matrix inversion algorithm (SMI), or any other least-squares algorithm. Refer to Fig. 2.6 and assume for the moment that the linear filters are time invariant. Then the transfer function from point d to the output is unity since the all-pole and all-zero filters cancel. This combined with the fact that the look-direction signal does not appear in the z_k inputs allows the look-direction signal to pass through the array undistorted. The transfer function from point z_k to the output is $\frac{B_k(z)}{1 + z^{-1}A(z)}$ and contains both poles and zeros. Instead of minimizing the overall array output, we minimize the signal appearing at the input to the all-pole filter. In the control literature this signal is often referred to as the equation error. By minimizing the equation error, we are able to adapt only all-zero filters, yet effectively realize a pole-zero filter response.

To convert the system of Fig. 2.6 into a system which uses constrained adaptation, observe that the transfer function from sensor k to the output is given by,

$$W_k(z) = \frac{\frac{1}{m}\left[1+ z^{-1}A(z)\right] - B_k(z) + B_{k-1}(z)}{1+ z^{-1}A(z)} \tag{2.18}$$

where $B_0(z) = B_m(z) = 0$. It turns out that eq. (2.18) can be equivalently written as,

$$W_k(z) = \frac{n_{k0} + z^{-1}N_k'(z)}{1+ z^{-1}A(z)} \tag{2.19}$$

where $\sum_{k=1}^{m} n_{k0} = 1$ and the $N_k'(z)$ are unconstrained. This leads to the alternate structural implementation shown in Fig. 2.7. The similarity between this system and the Frost system shown in Fig. 2.1 is apparent. Instead of constraining the sum of all the filters to be unity, only the sum of the first coefficients has been constrained. A look-direction signal will experience a

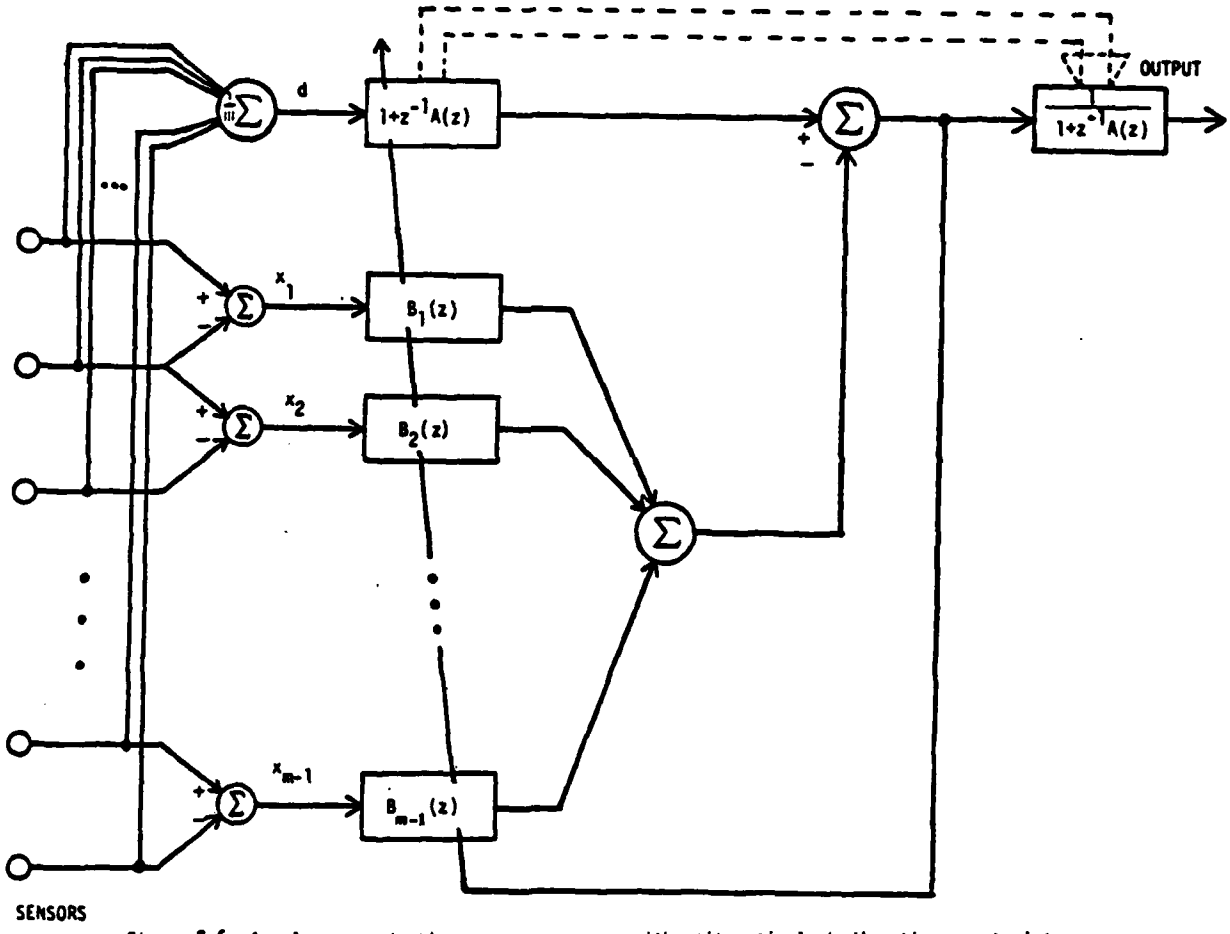


Figure 2.6 A pole-zero adaptive array processor with unity gain look direction constraint.

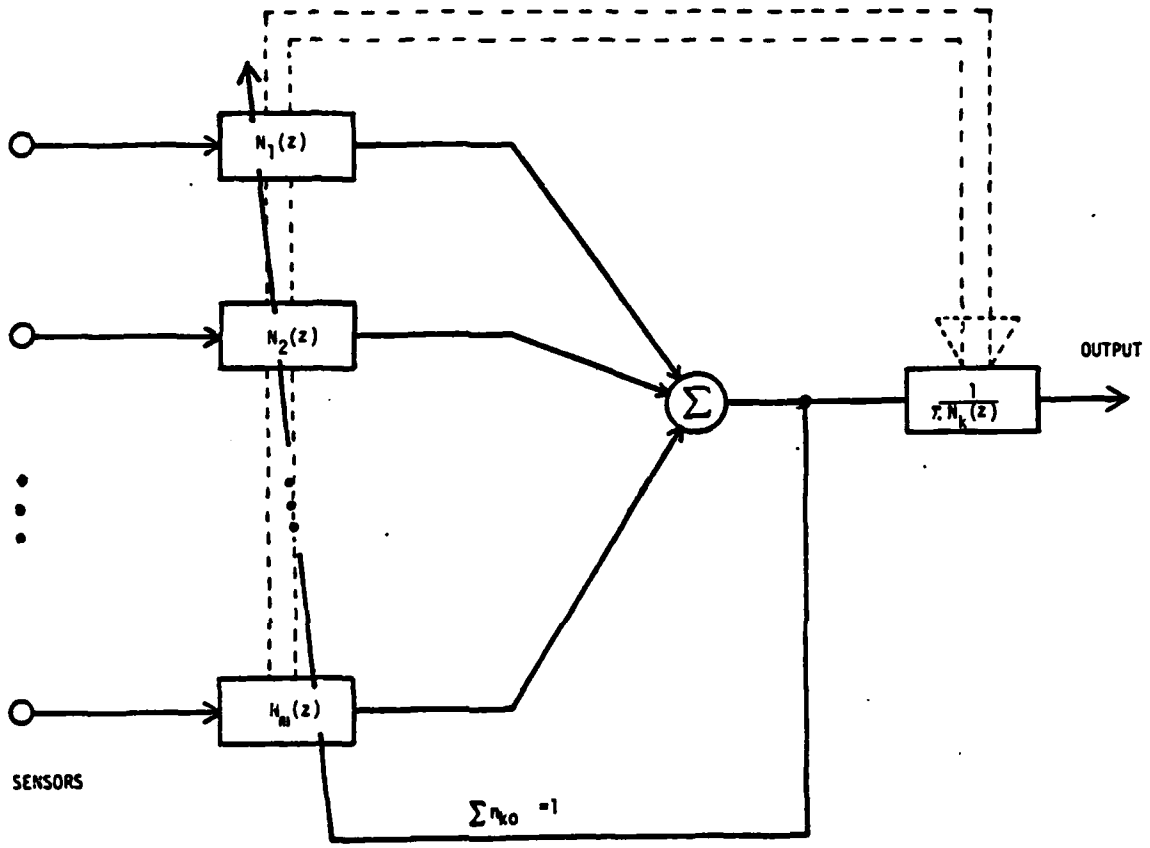


Figure 2.7 Alternate realization of a pole-zero array processor with unity gain look-direction constraint.

transfer function from input to the adaptation error point of

$$1 + z^{-1} \sum_{k=1}^{m} N_k'(z) \tag{2.20}$$

This response is inverted by the all-pole output filter to generate the array output and restore the look-direction constraint.

2.6 Simulation of the pole-zero Frost array processor

Figure 2.8 compares the frequency response of the optimal weight at each sensor with the frequency response of the converged 2 pole - 2 zero filter obtained using the proposed pole-zero adaptive array processor. As in the previous simulation, the interference source had a 100% relative bandwidth and an arrival angle 45 degrees off the look direction. Notice the improvement of this response over the all-zero response plotted in Fig. 2.4. For further comparison, Fig. 2.9 shows the broadband antenna pattern and the frequency response of the array in the signal and interference directions. The interference null is about 40 dB below the look-direction gain. This represents a 20 dB improvement over a conventional Frost system having the same number of degrees of freedom. Fig. 2.10 compares the nulling capabilities of the conventional Frost and the pole-zero Frost as a function of jammer arrival angle. Notice that as the jammer arrival angle approaches the look direction, the null depth goes to 0dB because of the unity gain look-direction constraint.

2.7 Stability of the inverse filter

One issue yet to be addressed is that of stability of the inverse filter. In some situations, it is possible that the roots of the polynomial associated with the inverse filter can move outside the unit circle. When this occurs, the inverse filter will become unstable and its output will begin to grow unboundedly. A solution with poles outside the unit circle represents a stable but noncausal filter and can only be realized by noncausally processing the input data. Since noncausal processing is not an available option when operating in realtime, pole polynomials with roots outside the unit circle must be avoided.

The method we propose for keeping all the pole polynomial roots inside the unit circle

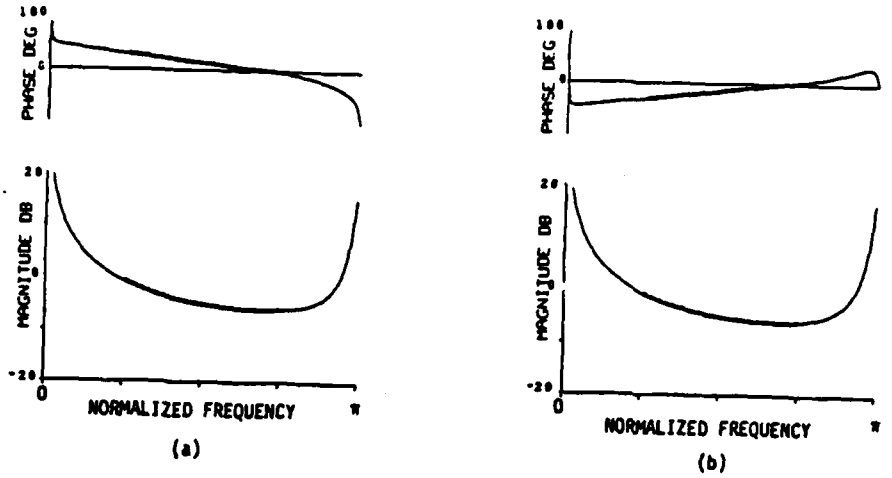


Figure 2.8 Comparison of frequency response of optimal array weights (cross-hatched marks) with a 2 pole - 2 zero approximation. (a) Sensor 1 (b) Sensor 2.

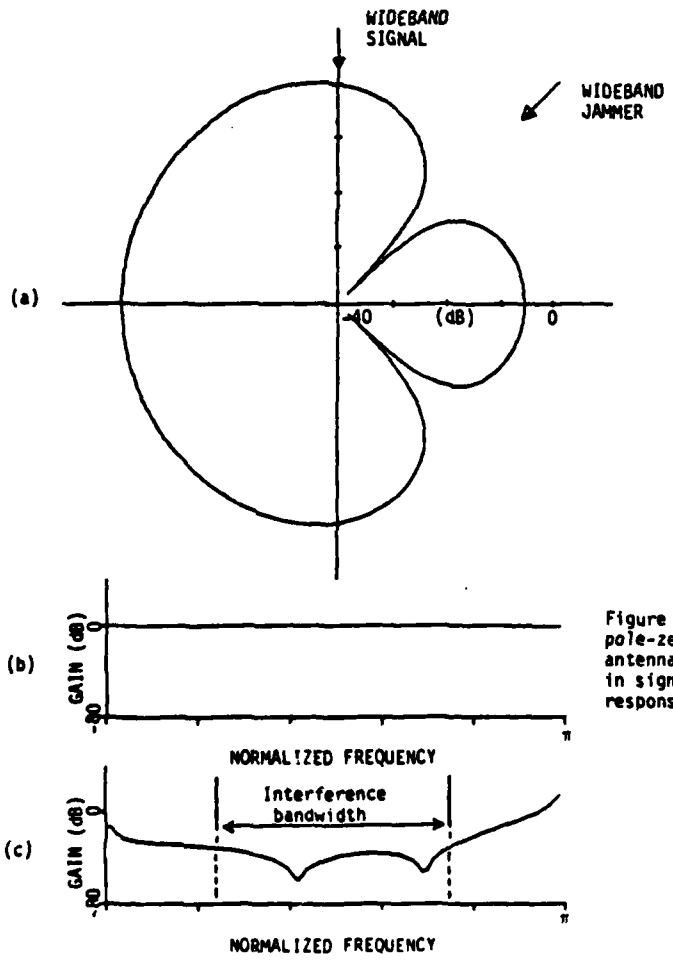


Figure 2.9 Nulling performance of a pole-zero Frost array. (a) Broadband antenna pattern, (b) frequency response in signal direction, (c) frequency response in interference direction.

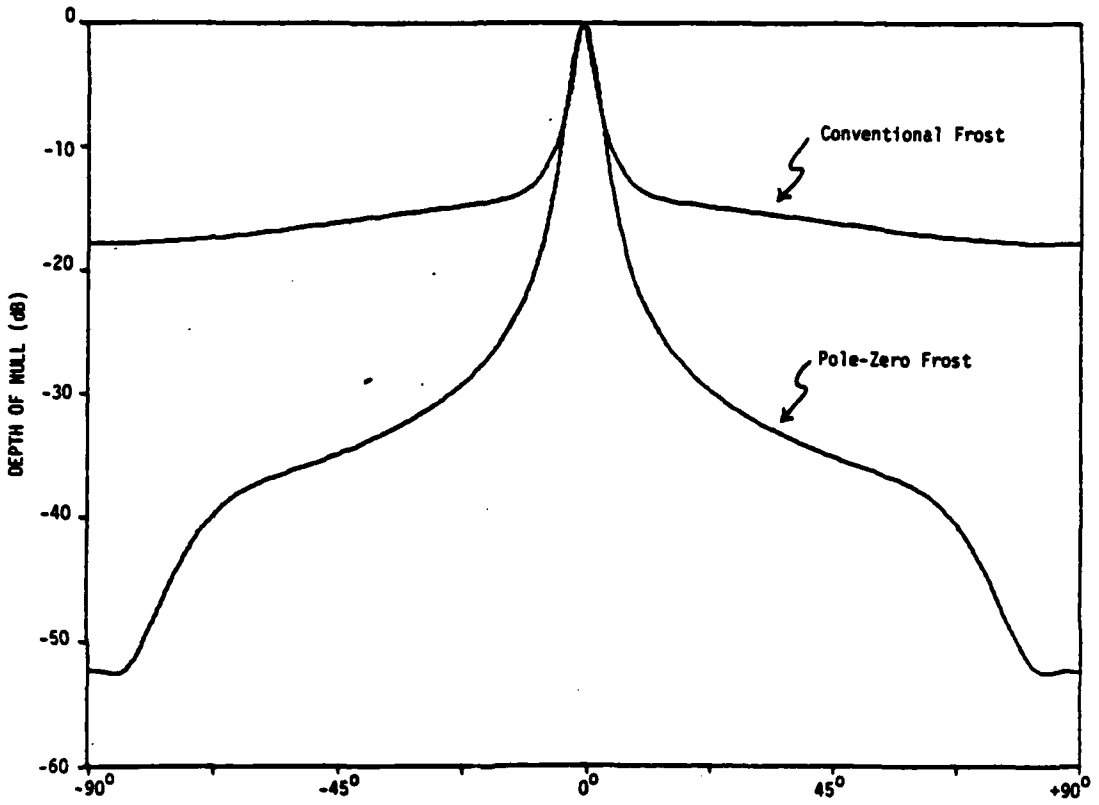


Figure 2.10 Nulling performance as a function of jammer angle.

involves the addition of a bulk delay in the desired response path of Fig. 2.6. Increasing the bulk delay causes the roots of the converged solution to lie closer to the origin. Choosing a delay too large will cause the resulting solution to lose quality and is not desirable. Choosing a delay too small will result in a noncausal solution and an unstable inverse filter. The idea is thus to choose the smallest delay which causes all the roots to lie within the unit circle.

In the example shown in Fig. 2.9, the converged solution was stable and bulk delay was not required. In fact, all of the two-element, single-jammer experiments we simulated turned out not to require any bulk delay. However, the three-element, two-jammer cases tried required one unit of bulk delay in order to keep the poles within the unit circle.

Currently, we are experimenting with several on-line methods for adaptively choosing the bulk delay. A brute force technique is to have a series of systems all operating on the same input data but each using a different delay. A simple criterion is used to choose the stable system having the smallest delay. Another technique is to use a single system with a variable delay which is incremented or decremented depending on the stability of the pole polynomial.

2.8 Look-direction signal bias

The Wiener solution of the conventional Frost array is not affected by the presence of a look-direction signal. This is readily apparent by referring to Fig. 2.3 and noting that the x-input does not contain look-direction signal. Therefore the look-direction signal cannot contribute to either the cross-correlation between the desired-response input d and the x-input or the autocorrelation of the x-input. The Wiener solution of the pole-zero Frost array is, however, affected by a look-direction signal. In this section, we demonstrate the extent of this effect and describe a modification of the pole-zero array which eliminates the dependence of the Wiener solution on the look-direction signal.

First, it should be noted that in situations where adaptation can be controlled so that it occurs only when the look-direction signal is not present, the problem of the look-direction signal disturbing the Wiener solution will not exist. This is the case in many radar or spread-spectrum

applications where the desired signal is pulsed or sequenced in a manner which is known to the receiver. In many continuous communication systems or seismic systems, the desired signal is continuous and an alternate solution must be found.

To demonstrate the effect of the look-direction signal on the Wiener solution of the pole-zero Frost array, refer to Fig. 2.6 and assume that the only source of signal comes from the look direction. In this condition, the x-input will be zero and the d-input will contain only signal. The $1+z^{-1}A(z)$ filter will act as a linear predictor and try to reduce the look-direction signal power. If the signal were narrowband, the predictor would form a notch filter at the signal frequency. This transfer function would theoretically be inverted by the inverse output filter to restore the look-direction signal at the array output so signal distortion is not the issue. The main concern is that degrees of freedom will be used in attempting to internally null the look-direction signal and thus cause a reduction of the interference nulling capabilities of the array. This effect is referred to as look-direction signal bias.

One method for alleviating the look-direction signal bias problem is shown in Fig. 2.11. The method is based on an idea proposed by Duvall for curing signal cancellation in adaptive arrays [7,8]. For simplicity, we have shown only a three-element example. The basic idea is to process the set of signals from the sensors in a manner which eliminates the look-direction signal but which keeps the phase relationships among the new set of signals the same as that prior to processing. The processed set of signals is fed into an adaptive "master" array where the Wiener solution will not be affected by the look-direction signal. The weights of the master array can be copied into the "slave" array which operates on the original signal set. Because of the preservation of phase, nulls formed using the slaved array will be in the same directions as those formed in the master.

To analyze the effect of preprocessing on the Wiener solution, we define an effective interference environment that incorporates the preprocessor into the interference sources. Each interference source will experience an angle-dependent frequency transformation when passed through the preprocessor. The frequency response of this transformation is determined by the subtraction of

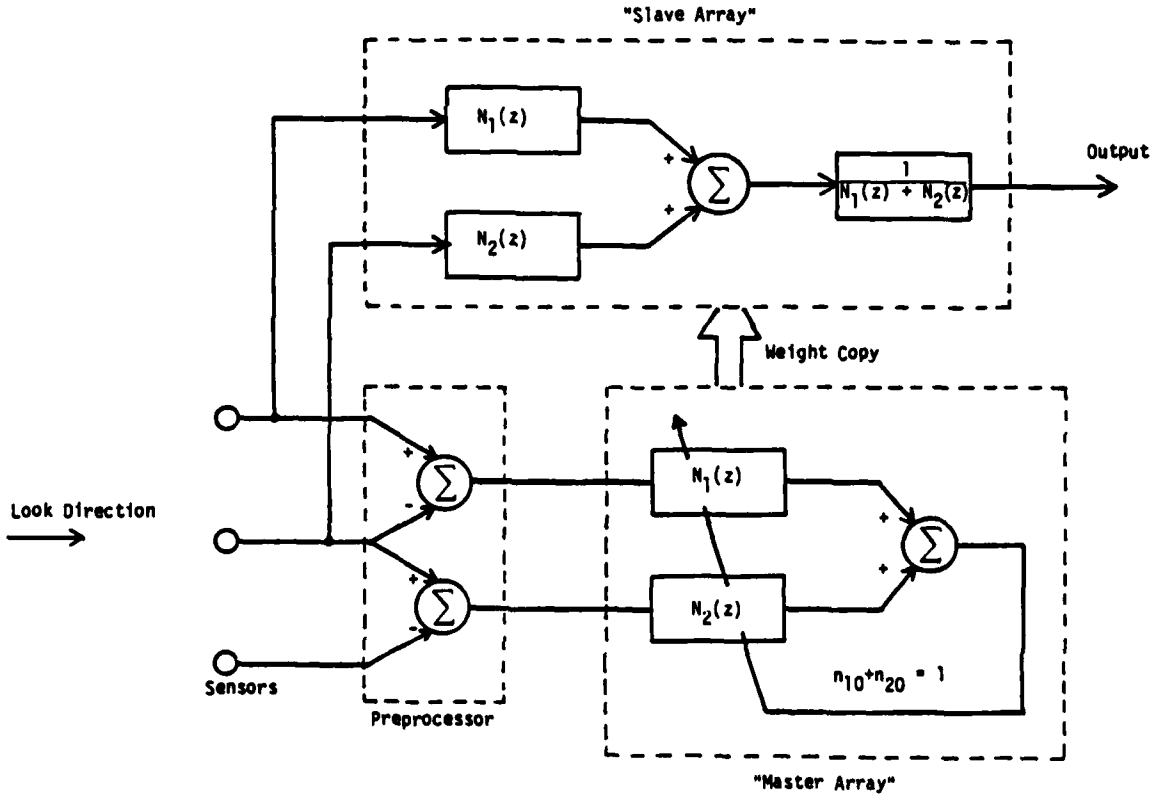


Figure 2.11 Method for eliminating bias in Wiener solution due to look-direction signal.

signals appearing at adjacent sensors.

Again we assume that the array of sensors is linear and uniformly spaced. A directional interference source will experience a delay, Δ, between two adjacent sensors that is determined by the arrival angle. The frequency response of the top subtractor in the preprocessor is given by

$$H(\omega\Delta) = 1 - e^{j\omega\Delta} \quad . \tag{2.21}$$

Proceeding down the array, each subtractor will give this same response along with an additional delay. The idea is to use the $H(\omega)$ transformation to modify the interference source and conceptually eliminate the preprocessor from the array. As the interference angle approaches the look direction, the magnitude of the transfer function goes to zero. Thus, the look-direction signal will be removed and an interference source appearing off the look direction will be somewhat attenuated and high-pass filtered. In this manner, the preprocessor affects the Wiener solution by modifying the weighting applied to the interference source.

III. THE SCALAR LEAST-SQUARES LATTICE FILTER

3.1 Introduction

Lattice filters were first introduced as a means of predicting speech signals. Work on linear prediction of speech using partial correlation techniques to whiten or decorrelate the input originated in 1969 [9]. Since then the range of applications for lattice filters has broadened to include joint-process estimation [10], signal enhancement [11] and equalization [12]. The least-squares lattice filter that we are now presenting was first used by Satorius and Pack [12] in the equalization context. The most interesting properties of the filter are presented here.

We begin by presenting the orthogonality principle of linear least-squares estimation and its application to the lattice predictor. Next, we present the lattice predictor first used for analyzing speech by decomposing it into a set of orthogonal components. We continue with the structure of the inverse of the analysis lattice, also known as the synthesis lattice in speech processing. We also give the necessary and sufficient conditions for the stability of that inverse filter. Next, we present the joint-process lattice for estimating a desired signal given an auxiliary one. Last, we present simulations of the parallel modeling of an unknown all-zero filter using the joint-process lattice.

3.2 Prediction and the orthogonality principle

As illustrated in Fig. 3.1, the forward prediction problem is that of estimating the next signal value $x(n)$ given M of its past values, from $x(n-M)$ to $x(n-1)$. Likewise, backward prediction involves estimating the past value $x(n-M-1)$ of the signal given M of its more immediate past values [13].

In the case of least-squares estimation, the orthogonality principle [14] states that by viewing the signal samples as elements of a vector space, the least-squares estimate of a sample is found by projecting it onto the subspace formed by the observed past M samples. As a result, not only is the current estimation error (also known as the residual or innovation) orthogonal to the past M samples but it is also orthogonal to all past M estimation errors. Fig. 3.2 illustrates

that orthogonality principle.

3.3 The Lattice Predictor

As shown in Fig. 3.3, the forward residuals f_m's are recursively formed by subtracting a linear combination of the past M samples of the input from the current sample $x(n)$. Likewise the backward residuals b_m's are formed by subtracting a linear combination of the same M samples from the oldest sample $x(n-M)$. The values of the linear combination are determined by the reflection coefficients h_m's and g_m's to minimize the weighted sum of f_m^2 and the weighted sum of b_m^2.

Two properties are worth mentioning at this point. First, from the orthogonality principle, each innovation f is orthogonal to any other innovation f. Likewise the backward innovations b's are orthogonal to each other.

The second property concerns the roots of the transfer functions from x to f_M and from x to b_M. We can consider those forward and backward prediction errors $f_m(n)$ and $b_m(n)$ to be the output of two filters $F_m(z)$ and $B_m(z)$ fed with the same input $x(n)$. Providing that $h_m = g_m$, (that occurs with stationary inputs) the polynomials F_m and B_m have roots that are inverse of each other [13] namely,

$$F_m(z_p) = 0 \iff B_m\left(\frac{1}{z_p}\right) = 0$$

Equivalently, we can conclude that $F_m(z)$ has the same coefficients as $B_m(z)$ but in reverse order.

3.4 Inverse of a lattice predictor

Thus far it seems that we have not really estimated the current sample $x(n)$ of a signal since we have assumed its availability to construct the lattice predictor. The answer to that argument has some historical aspects.

Part of the natural process of speech generation is often found to be adequately modeled by an all-pole filter $1/A(z)$ fed with white noise [13]. If we use that speech as the input to a lattice

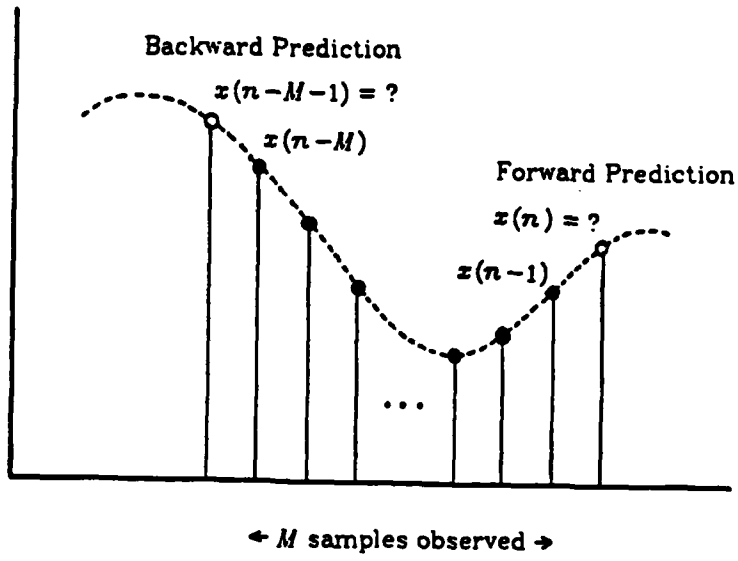


Fig. 3.1. Forward and backward predictions [after Markel and Gray].

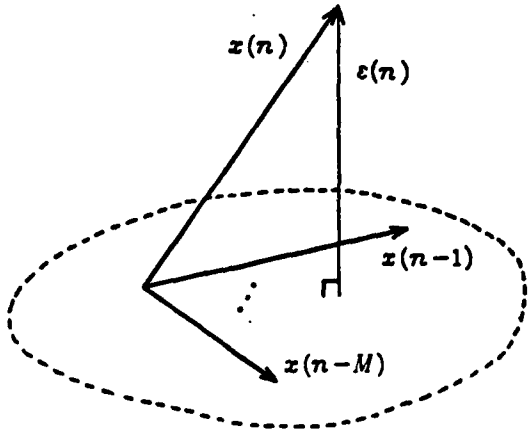


Fig. 3.2
$\varepsilon(n) \perp x(n-1), x(n-2), \cdots \Rightarrow \varepsilon(n) \perp \varepsilon(n-1), \cdots$
$\varepsilon(n)$ is the current estimation error or residual error. It is also known as the innovation as it represents the part of $x(n)$ that does not belong to the subspace spanned by its past values, i.e. the part that really is new.

predictor, the forward innovation f_M will be white noise providing that the order M is sufficient. Thus, the lattice transfer function from x to f_M is $\hat{A}(z)$, an estimate of the true filter denominator. The process of estimating $A(z)$ is known as speech analysis [13]. Eventually, we can synthesize part of the speech by feeding white noise into an inverse filter $1/\hat{A}(z)$ which is an estimate of the vocal tract transfer function as shown in Fig. 3.4.

Thus the complete prediction process involves taking the inverse of an all-zero filter. We now show that the inverse can be implemented in a structure that is very similar to that of a lattice.

We recall that the lattice recursions are

$$f_m(n) = f_{m-1}(n) + g_m\, b_{m-1}(n-1) \tag{3.1}$$
$$b_m(n) = h_m\ \ f_{m-1}(n) + \ b_{m-1}(n-1) \tag{3.2}$$

Since we are interested only in having f_{m-1} in terms of f_m, let us rewrite (3.1) leaving (3.2) unchanged

$$f_{m-1}(n) = f_m(n) - g_m\, b_{m-1}(n-1) \tag{3.3}$$
$$b_m(n) = h_m\, f_{m-1}(n) + \ b_{m-1}(n-1) \tag{3.2}$$

These recursions can be implemented as shown in Fig. 3.5 resulting in a structure having an interesting similarity with the analysis filter.

3.5 Necessary and sufficient stability conditions for the inverse

Although the experiment described in Fig. 3.4 clearly involves a stable $1/\hat{A}(z)$ filter, there are applications in which it is important to check the invertibility of the lattice predictor. For example in pole-zero filtering using the equation error, it is necessary to invert the filter in order to restore the true output error.

Necessary conditions

We recall that the lattice recursion can be written in terms of the polynomials $F(z)$ and $B(z)$ corresponding to the forward and backward time innovations, namely the forward

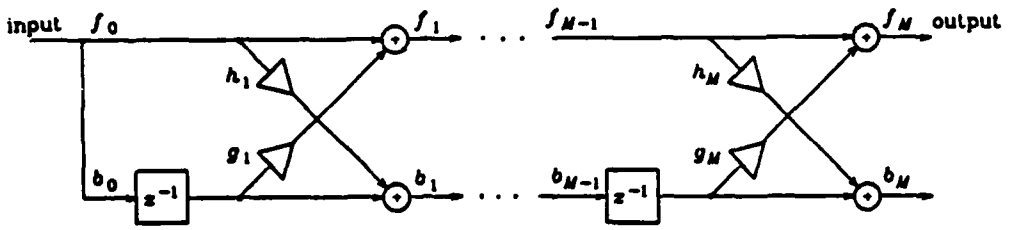


Fig. 3.3. Lattice predictor. The f_m's are the forward prediction errors and the b_m's are the backward prediction errors. The reflection coefficients h_m's and g_m's are updated to minimize both the weighted sum of f_m^2 and the weighted sum of b_m^2.

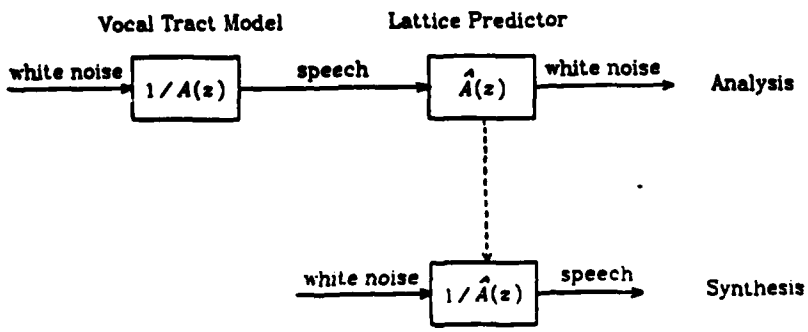


Fig. 3.4. Speech synthesis process. The speech generation model is first identified by an analysis lattice predictor. An inverse is then formed which yields the speech when fed with white noise.

innovation is updated as

$$F_m(z) = F_{m-1}(z) + g_m z^{-1} B_{m-1}(z) \tag{3.4}$$

Writing $F_m(z)$ in terms of its roots $\rho_{m,i}$ gives

$$\prod_{i=0}^{m} (1 - \rho_{m,i} z^{-1}) = F_{m-1}(z) + g_m z^{-1} B_{m-1}(z) \tag{3.5}$$

The initial conditions for the recursion are $F_0(z) = B_0(z) = 1$. Thus for $m \geq 1$,

$$\text{order } \left[F_{m-1}(z)\right] \leq m-1$$

and

$$\text{order } \left[B_{m-1}(z)\right] \leq m-1 \implies \text{order } \left[z^{-1} B_{m-1}(z)\right] \leq m$$

Thus the highest degree polynomial on the right-hand side of (3.4) is $z^{-1} B_{m-1}(z)$. Furthermore, the coefficient of the highest power of z^{-1} in $z^{-1} B_{m-1}(z)$ is 1. Equating the coefficients of the highest degree terms of both (3.4) and (3.5) gives

$$\prod_{i=1}^{m} (-\rho_{m,i}) = g_m \tag{3.6}$$

Evaluating the magnitude of both sides of (3.6) gives

$$\left|g_m\right| = \left|\prod_{i=1}^{m} (-\rho_{m,i})\right| \leq \prod_{i=1}^{m} \left|\rho_{m,i}\right|$$

Thus if the roots $\rho_{m,i}$ have magnitude less than 1, the product of those magnitudes is less than 1 implying that the magnitude of the reflection coefficient g_m be less than 1 too. A necessary condition for the roots of the end-polynomial $F_M(z)$ to be within the unit circle is therefore

$$\frac{1}{F_m(z)} \text{ stable } \implies \left|g_m\right| < 1$$

Sufficient conditions

Using a more complete analysis, Markel and Gray [13] were able to show that for *equal reflection coefficients* a necessary and sufficient condition for the stability of $1/F_M(z)$ is that the

reflection coefficients for all stages be of magnitude less than 1, namely

$$\frac{1}{F_M(z)} \text{ stable } <=> \left|k_m\right| < 1, \quad m = 1,2,...,M$$

where

$$k_m = h_m = g_m$$

That requirement of equal reflection coefficients is not necessarily stringent since in practice an equivalent lattice predictor meeting that condition is computed before stability can be checked (see section V).

3.6 Joint-process lattice

The situation often arises in signal processing when we want to filter one signal to identify another signal or to remove any correlation of it with the former one. The filtering of x to get an estimate of d can be done by the joint-process lattice of Fig. 3.6. The backward innovations b_m are used as a basis on which the desired signal d is decomposed. The gains ν_m's are determined so as to minimize the sum of the ϵ_m^2. Also because of the orthogonality, it is possible to first estimate d with $\nu_0 b_0$ yielding an error ϵ_0 which is then estimated by $\nu_1 b_1$ yielding an error ϵ_1 etc.

Last, because of the whitening that is taking place, we expect the joint-process lattice to exhibit a convergence rate that is independent of the x-input eigenvalues.

3.7 Simulations

The simulation context is the parallel modeling of an unknown all-zero filter, $B(z) = 1 + 2z^{-1} + 4z^{-2} + 2z^{-3} + z^{-4}$. The joint-process lattice is fed with the same input as the unknown filter and the output of the lattice is the desired signal of the lattice.

To minimize any initial conditions effect, we do not look at the convergence of the lattice from its start-up. Rather, we wait for the weights to converge then inject an impulse as a disturbance at the output of the unknown filter. Our criterion is therefore the time it takes to reject that disturbance.

Furthermore, to truly evaluate the convergence time, the simulations are noise-free, yielding

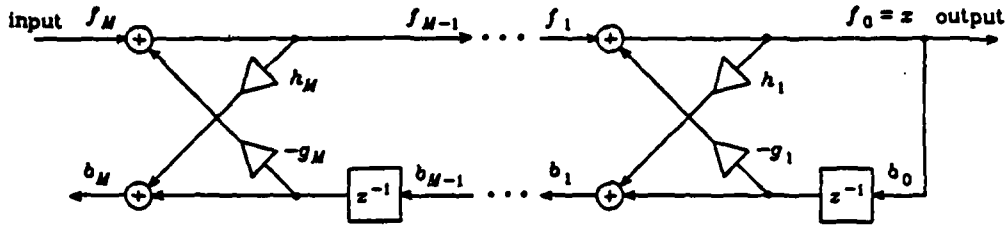


Fig.3.5 Implementation of an inverse filter given the reflection coefficients h_m and g_m of the analysis filter. Note that only the upper path of the analysis filter is reversed. Also, note the sign inversion on the g_m's.

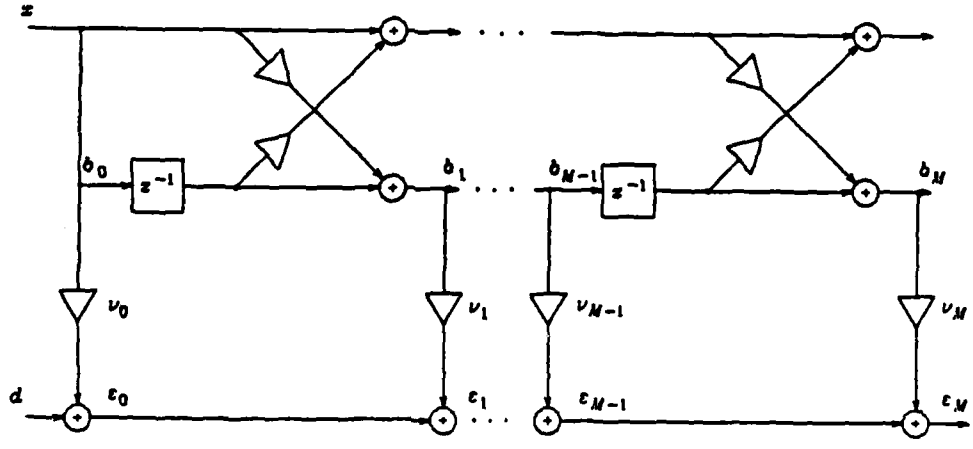
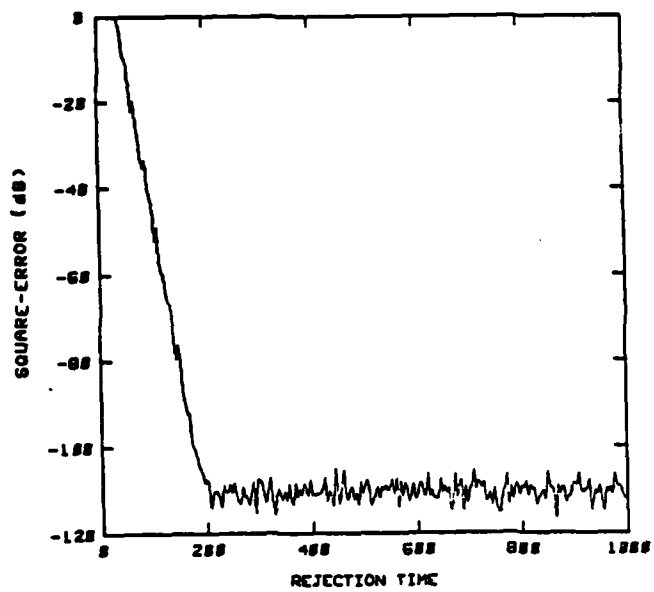


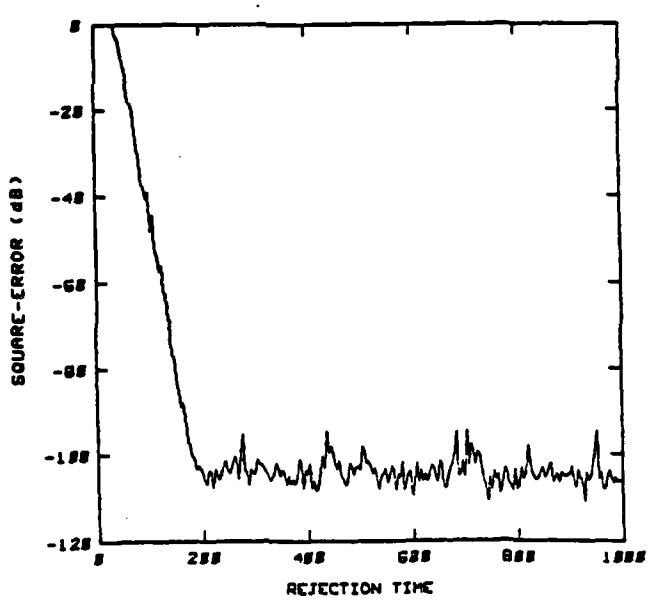
Fig. 3.6
Joint process lattice to estimate one signal given another.

a minimum mean-square error of the order of -100 dB. It is understood that in actual applications the noise level is much higher. For evaluation purposes however, only a noise-free environment makes the algorithm exhibit its full convergence potential.

Fig. 3.7(a) and 3.7(b) show the mean-square error of the joint-process lattice for an input eigenvalue ratio of 1 (white noise) and 10 respectively. As expected, the convergence time remains essentially constant.



(a)



(b)

Fig.3.7 Disturbance rejection curve of scalar joint-process lattice for an input eigenvalue ratio of
(a) 1
(b) 10

IV. POLE-ZERO FILTERING USING A TWO-CHANNEL LATTICE PREDICTOR

4.1 Introduction

In this section, we present a novel approach to adaptive pole-zero filtering. We show that the equation-error method can be implemented using a two-channel (2-C) lattice predictor. We begin by presenting the 2-C lattice predictor as an extension of the scalar predictor discussed in section III. Next, we reformulate the equation-error approach introduced in previous reports in terms of linear prediction and then present the equation-error lattice predictor used for pole-zero filtering. Last, a simulation involving the identification of an unknown pole-zero filter is presented demonstrating the relative insensitivity of the 2-C lattice predictor to an eigenvalue disparity.

4.2 The 2-C lattice predictor

The scalar lattice predictor presented in section III can be generalized to a multi-dimensional lattice predictor when the process to be whitened is itself multi-dimensional. The algorithm is detailed in [15]. For the moment, we restrict ourselves to considering only the 2-C case. The multi-dimensional case is considered in section V for multi-element antenna array processing.

Given a vector

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

the lattice predictor constructs a sequence of forward and backward innovation vectors

$$f_m = \begin{bmatrix} f_m^{x_1} \\ f_m^{x_2} \end{bmatrix} \text{ and } b_m = \begin{bmatrix} b_m^{x_1} \\ b_m^{x_2} \end{bmatrix}$$

In particular, $f_M^{x_1}$ (see Fig. 4.1) is the forward prediction error of x_1 gi the past M values of x_1 *and* x_2, namely,

$$f_M^{x_1} = x_1(n) - \hat{x}_1(n) \,\Big|\, x_1(n-1),\ldots, x_1(n-M), x_2(n-1),\ldots, x_2(n-M)$$

Recognizing the forward prediction error as such will allow us to use the 2-C lattice predictor in

pole-zero array processing.

4.3 Reformulation of the equation error approach

The equation error method has been discussed in detail in reference [16]. The objective is to filter the auxiliary input x with an autoregressive and moving average (ARMA) filter, $B(z)/(1 + z^{-1}A(z))$, i.e. using poles and zeros so that the output error is minimized (see Fig. 4.2(a)). As shown in the previous reports, the $A(z)$ and $B(z)$ filters can equivalently estimated by minimizing the equation error between d filtered through $(1 + z^{-1}A(z))$ and x filtered through $B(z)$ (see Fig. 4.2(b)).

As shown in Fig. 4.2(c), the equation error method can be thought of as the minimization of the forward prediction error of d given the past values of d, the current value of x and the past values of x. This is essentially what a 2-C lattice predictor does.

4.4 Equation-error lattice predictor

We are changing the seemingly scalar problem of pole-zero filtering into a multivariable problem by embedding the primary and auxiliary signals into a vector.

We recall that in the 2-C lattice predictor, the prediction of any element of the input vector $\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ is based upon past values of both elements x_1 or x_2 of that vector. The equation error approach requires that we also base the prediction of d on the current value of x. Therefore the d-input needs to be delayed by one in order to obtain a correct prediction as shown in Fig. 4.3.

As in the scalar case, we expect this novel adaptive scheme to be relatively insensitive to any disparity in the eigenvalues of the input vector. Such a property might be very useful in some instances. In actual system identification for example, the auxiliary signal x (the input to the unknown plant) is seldom white thus has an eigenvalue spread. But even if it is, the primary signal d is nothing but colored since it is the plant output.

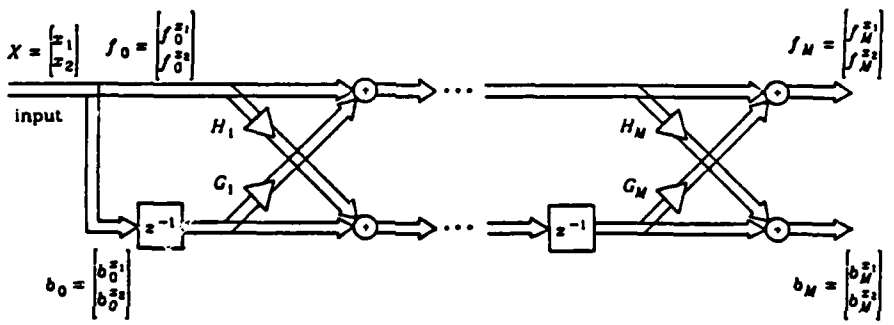


Fig. 4.1. The 2-C lattice predictor, a straightforward extension of the scalar predictor. The filter whitens a vectorial process X.

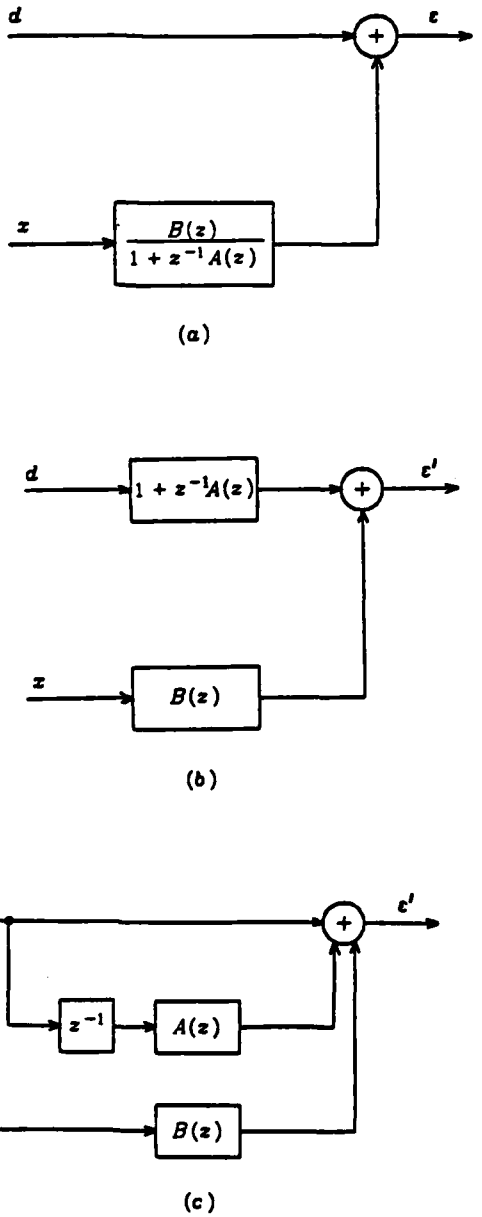


Fig. 4.2
(a) Pole-zero filtering to minimize the output error ε.
(b) $A(z)$ and $B(z)$ can be identified by minimizing the equivalent equation error ε'.
(c) Identification of $A(z)$ and $B(z)$ by minimizing the prediction error which is the same as ε'.

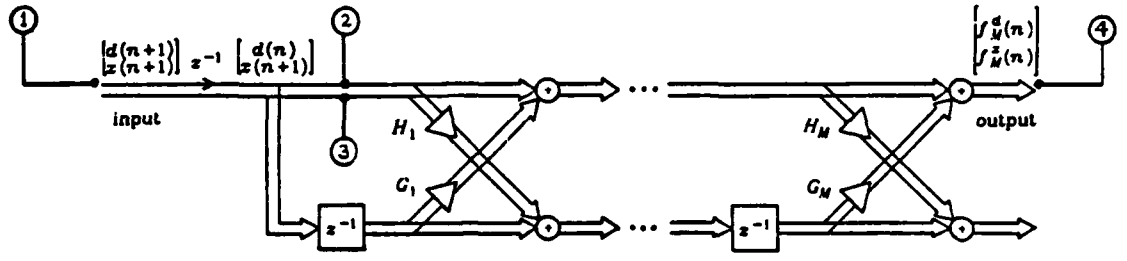


Fig. 4.3 Equation error approach using a 2-C lattice predictor of order M.

The output of the predictor is

$$f_M^d(n) = d(n) - \hat{d}(n) \,\big|\, d(n-1), \ldots, d(n-M), x(n), \ldots, x(n-M+1)$$

The transfer function from 1 to 4 and from 3 to 4 identify the polynomials $A(z)$ and $B(z)$ namely,

$$H_{14}(z) = z^{-1}(1 + z^{-1}A(z)) \quad \text{and} \quad H_{24}(z) = z^{-1}B(z).$$

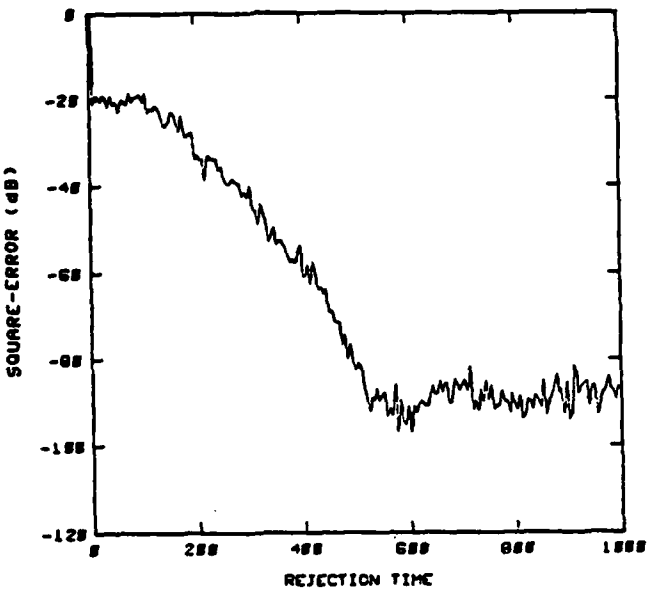
4.5 Simulations

The simulation context is the modeling of an unknown 4th order Butterworth filter, a filter with both poles and zeros. The predictor x-input is the input to the Butterworth filter. The predictor d-input is the output of the Butterworth filter.

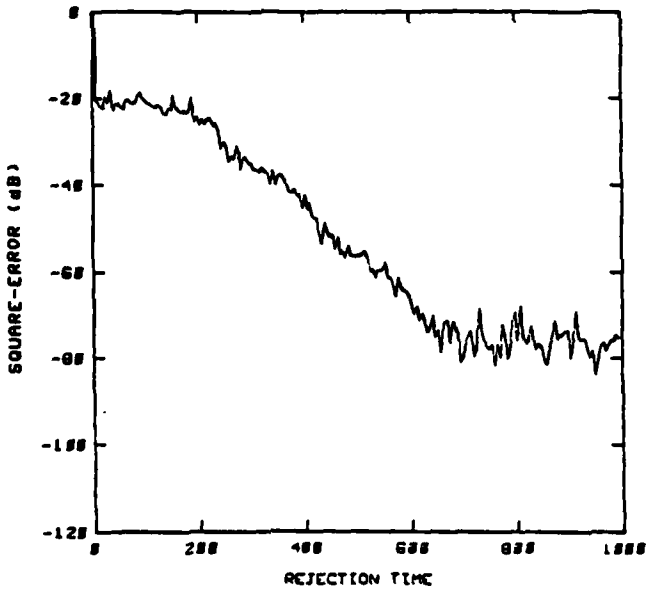
As in section III, we only look at the time to reject an impulse disturbance at the output of the unknown filter. By doing this, we minimize the effect of initial conditions. Also, the simulation is noise-free in order to evaluate the full convergence potential of the algorithm.

Fig. 4.4(a) and 4.4(b) show the mean square prediction error of the 2-C lattice for an x-input eigenvalue ratio of 1 (white noise) and 10 respectively.

The convergence time is slightly longer for a ratio of 10 than for pure white noise (650 iterations vs. 550) but it remains very rapid.



(a)



(b)

Fig. 4.4. Disturbance rejection curve of 2-C pole-zero lattice predictor for an x-input eigenvalue ratio of
(a) 1
(b) 10

V. TWO-ELEMENT ANTENNA PROCESSING USING A POLE-ZERO LATTICE

5.1 Introduction

The equation error approach as applied to antenna array processing, requires the formation of the inverse filter $1/[1 + z^{-1}A(z)]$ in addition to the estimation of $A(z)$ and $B(z)$. In this section, we first present a way of forming that inverse and checking its stability. Next, we show an implementation of the pole-zero lattice predictor in the adaptive array configuration presented in section II of this annual report. By removing the constraints in the antenna adaptive filters, the equation error array processing structure is suitable for implementation using the rapidly converging lattice predictor we are considering. Last, we present the simulation results for a wideband jammer cancellation experiment.

5.2 Inverse of $1 + z^{-1}A(z)$

To restore the true output error, we need to be able to invert the $1 + z^{-1}A(z)$ polynomial, i.e. the transfer function from 2 to 4 in Fig. 4.1. This requires that $H_{24}(z) = 1 + z^{-1}A(z)$ has a stable inverse. But we need to invert only that transfer function unlike in section II where we inverted the complete upper path of the forward innovations. Doing this also inverts $H_{34}(z)$ and thus requires the latter to be minimum-phase. Such a double constraint is certainly sufficient to ensure the stability of $1/(1 + z^{-1}A(z))$ but is not necessary.

A computationally more complex method for checking the stability of the polynomial $H_{24}(z)$ is to convert it to a scalar lattice form using what is known as the "step-down" procedure [13]. Then we need only recall that the necessary and sufficient conditions for $H_{24}(z)$ to have roots within the unit circle is that all reflection coefficients of its lattice implementation be of magnitude less than one.

5.3 Pole-zero adaptive lattice for array processing

The array processor shown in Fig. 5.1 is a pole-zero extension of the Griffiths and Jim generalized sidelobe canceller [6]. It is presented in detail in section 2.3 of this report. Any signal

coming from the look direction is constrained to pass through the array undistorted. The preprocessing of the sensor outputs allows the constraints to be removed from the adaptive algorithm and allows us to use the same lattice predictor as we used in the system identification experiment.

As suggested previously, a scalar lattice predictor equivalent to $1 + z^{-1}A(z)$ needs to be computed and the magnitude of the reflection coefficients checked before an inverse is actually implemented.

5.4 Simulations

Fig. 5.2 shows the simulation results of the pole-zero adaptive lattice used for a 2-element antenna. Those results are to be compared with the simulations presented in section 2.6 of this report.

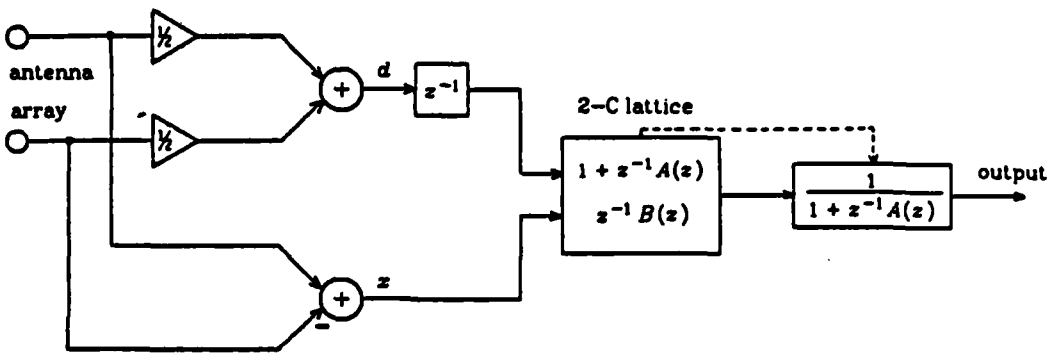
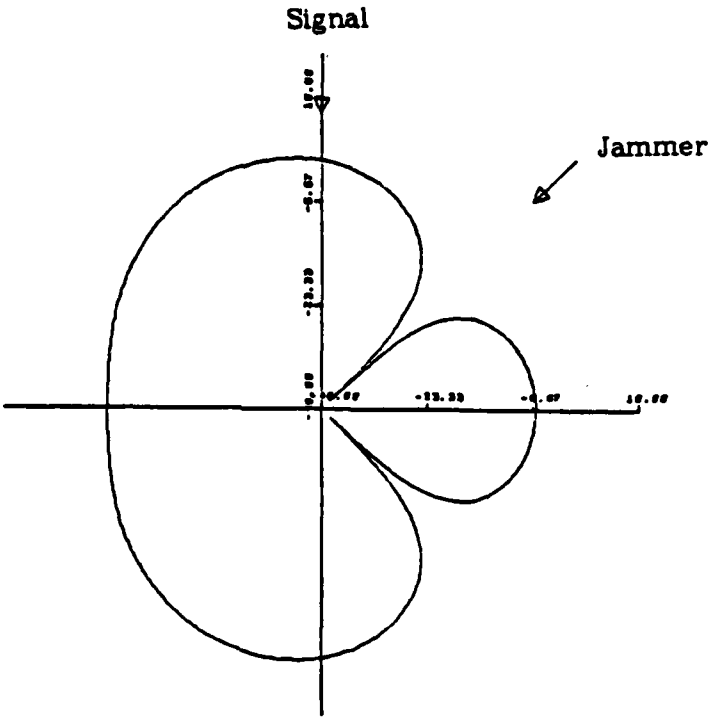
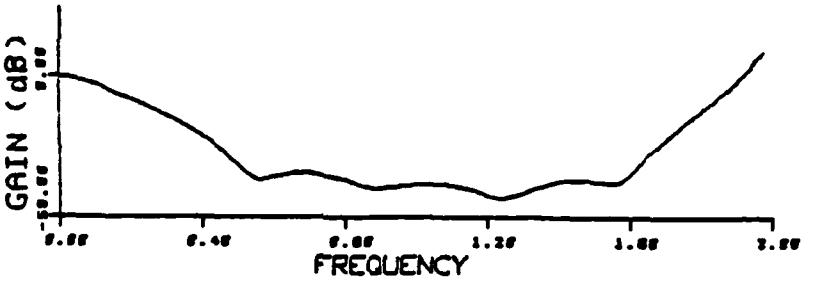


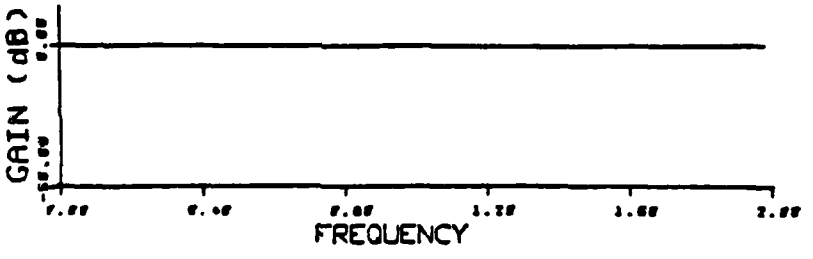
Fig.5.1 Array processing using a pole-zero adaptive lattice. Note that the output is delayed by one time unit due to the delay in the primary input path.



(a)



(b)



(c)

Fig.5.2 2-C lattice antenna array.
(a) Broadband antenna sensitivity pattern.
(b) Frequency response in the jammer direction.
(c) Frequency response in the signal direction.

VI. MULTI-ELEMENT ARRAY PROCESSING USING A POLE-ZERO LATTICE

6.1 Introduction

The use of the equation error approach in multi-element array processing was addressed in section II of this report. It is summarized here for convenience and in order to demonstrate the connection to dimensional lattice prediction.

As shown in Fig. 6.1(a), the pole-zero filters $B_p(z)/(1 + z^{-1}A_p(z))$ need to be adjusted to minimize the output error which is similar to minimizing the equation error shown in Fig.6.1(b). Equation error minimization is equivalent to minimizing the prediction error of d given past values of d, the current values of x_p's and the past values of x_p's (see Fig. 6.1(c)).

6.2 Implementation of the multi-channel pole-zero lattice

Similarly to what was done in the 2-C case, multi-channel pole-zero filtering can be embedded into multi-channel prediction. The implementation of such a predictor in lattice form is shown in Fig. 6.2. Again, we require that the primary input path have a unit-delay in order to base the estimation of the primary signal in the current values of the auxiliary signals as well.

6.3 Limitations of the multi-channel pole-zero lattice

Although theoretically appealing, the multi-dimensional pole-zero lattice requires a large number of computations. In particular, there are $2\,[(P + 1) \times (P + 1)]$ matrix inversions each time a stage is updated where $P + 1$ is the total number of lattice inputs.

The forward and backward innovations are updated from order $m-1$ to order m as follows [15]

$$f_m(n) = f_{m-1}(n) + G_m(n-1)\, b_{m-1}(n-1)$$
$$b_m(n) = H_m(n-1)\, f_{m-1}(n) + b_{m-1}(n-1)$$

where the crossover gain matrices G_m and H_m are computed by

$$G_m(n-1) = K_m^T(n-1)\, E_{m-1}^{-b}(n-2)$$
$$H_m(n-1) = K_m(n-1)\, E_{m-1}^{-f}(n-1)$$

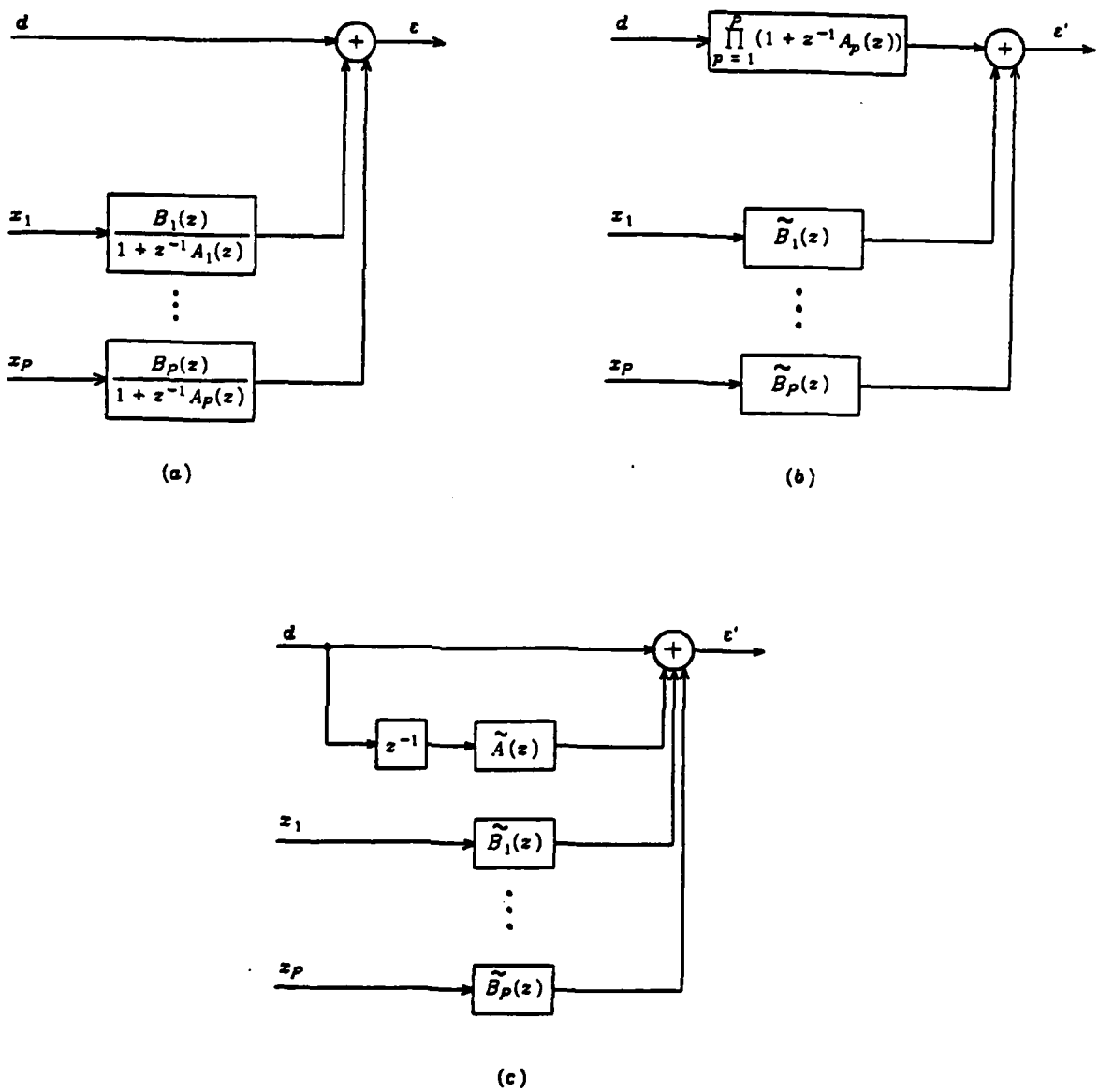


Fig 6.1 Equation error in terms of prediction error
(a) Multi-element array filtering with poles and zeros to minimize the output error ε.
(b) Processing using only zeros to equivalently minimize the equation error ε'.
(c) ε' is in fact the prediction error of d given past d's, the current x_p's and the past x_p's. The filter $A(z)$ is defined as

$$\tilde{A}(z) = -1 + \prod_{p=1}^{P} (1 + z^{-1} A_p(z))$$

The filters $\tilde{B}_p(z)$ are defined as

$$\tilde{B}_p(z) = B_p(z) \prod_{\substack{q=1 \\ q \neq p}}^{P} (1 + z^{-1} A_p(z))$$

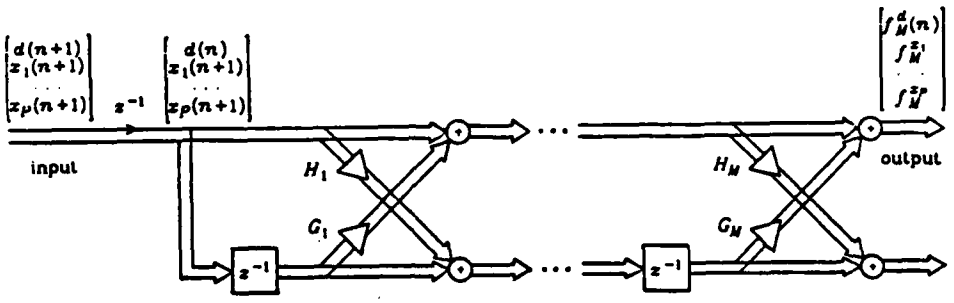


Fig. 6.2. Equation error approach using a multi-C lattice predictor.

where the $[(P + 1) \times (P + 1)]$ matrices E_{m-1}^{-b} and E_{m-1}^{-f} are the inverses of the minimized sums of the backward and forward prediction errors respectively. Those inversions take on the order of P^2 operations which make the crossover gain computations on the order of P^3 operations. The overall lattice recursion is then dominated by a count on the order of $M \times P^3$ operations where M is the number of stages. If P is small with respect to M that computational disadvantage may not be too crucial but if P is large or simply close to M, any improvement in convergence speed brought by the lattice would be heavily paid in algorithmic complexity.

VII. CONCLUSIONS

In this report, we have presented a new wideband adaptive array processing structure Using a derivation of the optimal wideband array weighting function, we conjectured that filters possessing both poles and zeros should be capable of improving the performance of the adaptive array. Using the "equation-error" approach, we developed an adaptive method for adjusting the poles and zeros of the array filters. Through simulations, we showed that the equation error approach indeed does improve the nulling capabilities of the array.

Also in this report, we discussed scalar and multi-channel lattice filters. Applications of the lattice to linear prediction, joint-process estimation, general equation-error pole-zero filtering, and pole-zero array processing were discussed. The major motivation for the use of lattice filters is their fast convergence properties. Simulations were presented which confirmed this fact.

The information presented in this report is by no means the final word on equation-error array processing. Several issues need further consideration a few of which we are listed below. Maintaining stability of the inverse output filter is a major issue. Techniques to easily check this filter possibly using the lattice approach need further investigation. A technique for adaptively choosing the minimum bulk delay which stabilizes the pole polynomial needs to be further developed and analyzed. Additional experiments should be conducted with the method presented for elimination of look-direction signal bias. The algorithm for implementing a multi-channel lattice filter should be modified in order to reduce the computational complexity from being proportional to the cube of the number of channels down to the square of the number of channels.

REFERENCES

[1] O. L. Frost, III, "An algorithm for linearly constrained adaptive array processing," Proceedings of the IEEE, Vol. 60, No. 8, pp 926-935, Aug. 1972.

[2] R. Monzingo and T. Miller, *Introduction to Adaptive Arrays,* John Wiley and Sons, New York, 1980.

[3] W. Rodgers and R. Compton, Jr., "Adaptive array bandwidth with tapped delay-line processing," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-15, No. 1, pp. 21-27, Jan. 1979.

[4] J. Mayhan and A. Simmons, "Wide-band adaptive antenna nulling using tapped delay lines," IEEE Transactions on Antennas and Propagation, Vol. AP-29, No. 6, pp 923-936, Nov. 1981.

[5] S. Applebaum and D. Chapman, "Adaptive arrays with main beam constraints," IEEE Transactions on Antennas and Propagation, Vol. AP-24, No. 5, pp. 650-662, Sept. 1976.

[6] L. J. Griffiths, C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming", IEEE Transactions on Antennas and Propagation, Vol. AP-30, No. 1, Jan. 1982.

[7] B. Widrow, K. Duvall, R. Gooch, and B. Newman, "Signal cancellation phenomena in adaptive antennas: causes and cures," IEEE Transactions on Antennas and Propagation, Vol. AP-30, No. 3, pp 469-478, May 1982.

[8] K. Duvall, "Signal cancellation in adaptive antennas," Ph.D. dissertation, Dept. of Elec. Eng., Stanford University, Sept. 1982

[9] F. Itakura, S. Saito, "Speech analysis-synthesis system based on the partial autocorrelation coefficient," Acoust. Soc. of Japan Meeting, 1969.

[10] M. Morf, D. Lee, "Recursive least squares ladder forms for fast parameter tracking," Proc. IEEE Conf. on Decision and Control, pp. 1362-1367, January 1979.

[11] V. U. Reddy, B. Egardt, T. Kailath, "Optimal lattice-form adaptive line enhancer for a sinusoidal signal in broad-band noise," IEEE Transactions on Circuits and Systems, Vol. CAS-28, No. 6, June 1981.

[12] E. H. Satorius, J. D. Pack, "Application of least squares lattice algorithms to adaptive equalization," IEEE Transactions on Communications, Vol. COM-29, No.2, February 1981.

[13] J. D. Markel, A. H. Gray, *Linear Prediction of Speech*, Springer-Verlag, New York, 1976.

[14] B. D. O. Anderson, J. B. Moore, *Optimal Filtering*, Prentice Hall, New Jersey, 1979.

[15] M. S. Mueller, "Least squares algorithms for adaptive equalizers", Bell System Technical Journal, Vol. 60, No. 8, October 1981.

[16] B. Widrow et al., "Research on algorithms for adaptive array antennas", Annual Interim Report, Rome Air Development Center, February 1981.


2-8
DT